CAIRO UNIVERSITY
INSTITUTE OF STATISTICAL STUDIES & RESEARCH
JUNE 2015
TIME: 3 HOURS

COMPUTER SCIENCE DEPARTMENT
DIPLOMA OF INFORMATION SCIENCES
ANALYSIS AND DESIGN ALGORITHM (CS512)
DR. Mahmoud Abdel Moneim Mahmoud

ANSWER THE FOLLOWING QUESTIONS

Question 1: (15 marks)

Consider the following pseudo code for the algorithm Gnome Sort:

```
function GnomeSort(Array A)
I ← 1
while i ≤ n do
if (i == 1) or (A[i - 1] ≤ A[i]) then
i++
else
Exchange A[i] ↔ A[i - 1]
i--
end if
end while
end function
```

a) Analyze the best- and worst-time complexity of gnome sort

b) Prove that the algorithm is correct

Question2: (10 marks)

a) For every given f(n) and g(n) prove that $f(n) = \Theta(g(n))$

1- $g(n) = n^3$, $f(n) = 3n^3 + n^2 + n$

2- $g(n) = 2^n$, $f(n) = 2^{n+1}$

b) For every given f(n) and g(n) prove that $f(n) = o(g(n))$ or $f(n) = w(g(n))$

1- $f(n) = n^3$, $g(n) = n^2$

2- $f(n) = \log(n)$, $g(n) = \log_2(n)$

C) Solve the following recurrence using the Master method.

1- $T(n) = 7T(n/2) + n^2$

Question 3: (10 marks)

Use the **divide-and-conquer** approach to write an algorithm **that finds the largest item in a list of n- items.** Using the following recurrence relation

$$T(n) = \begin{cases} 2T\left(\frac{n}{2}\right) + \Theta(1) & n > 1 \\ \Theta(1) & n = 1 \end{cases}$$

Question4: (15 marks)

The optimal local sequence alignment problem is about determining similar regions between two protein sequences. For example for the two sequences
Sequence 1 = ACACACTA
Sequence 2 = AGCACACA
We would get:
Sequence 1 = A - CACACTA
Sequence 2 = AGCACAC –A
All symbols in the two sequences have to be in the alignment and in the same order as initially. One possible dynamic programming algorithm compares segments of all possible lengths and optimizes the *similarity measure* by building a Matrix alignment M. The cells indicate the best score of alignment so far.

The algorithm builds a 2D matrix M whereby the cell values are computed recursively using an additional weight function $w(a_i,b_j)$ such that:

$$w(a_i,b_j) = \begin{cases} 2 & \text{if } a_i = b_j \\ -1 & \text{if } a_i \neq b_j \end{cases}$$

Inspired by the optimal longest common subsequence problem studied in class, and using the weight function, provide a recursive formulation to compute the M matrix.

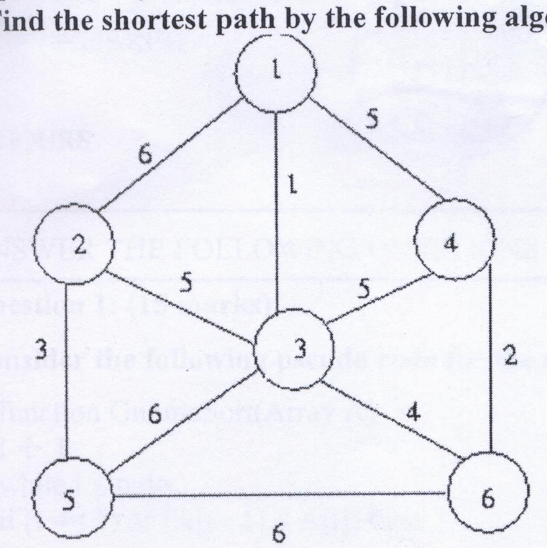
Hint: add a first row and a first column of 0.
M(i,0) = 0 for all i
M(0,j) = 0 for all j.

Question 5: (5 marks)

Find Huffman codes and compression ratio (C.R.) for the following Table, assuming that uncompressed representation takes 8-bit per character and assume that size of Huffman table is not part of the compressed size.

Char	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Frequency	**90**	**60**	**50**	**20**	**12**	**8**	**7**	**3**

Question 6 (15 marks)
Find the shortest path by the following algorithms if the source node is (1)



1- **Kruskal algorithm**
2- **Prime algorithm**
3- **Dijkstra algorithm**

Good Luck ☺
Dr. Mahmoud Abdel Moneim Mahmoud